Comp. & Maths. with Appls. Vol. 9, No. 1, pp. 131–147, 1983
Printed in Great Britain

0097-4943/83/010131-17$03.00/0
Pergamon Press Ltd.

UNDERSTANDING NOVEL LANGUAGE†

GERALD F. DEJONG and DAVID L. WALTZ
Coordinated Science Laboratory and Electrical Engineering Department, University of Illinois, Urbana, IL 61801, U.S.A.

Abstract—In this article we treat in some detail the problem of designing mechanisms that will allow us to deal with two types of novel language: (1) text requiring scheme learning; and (2) the understanding of novel metaphorical use of verbs. Schema learning is addressed by four types of processes: schema composition, secondary effect elevation, schema alteration, and volitionalization. The processing of novel metaphors depends on a decompositional analysis of verbs into "event shape diagrams," along with a matching process that uses semantic marker-like information, to construct novel meaning structures. The examples we describe have been chosen to be types that occur commonly, so that rules that we need to understand them can also be used to understand a much wider range of novel language.

1. INTRODUCTION

Natural language understanding systems are interesting to the extent that they understand material that they were never explicitly programmed to handle. A system such as ELIZA[1] or PARRY[2], which operates primarily by pattern matching, is less interesting than a system which has a set of general rules that can be used to generate a meaning representation for unanticipated inputs. There are a wide variety of types of unanticipated input. Some examples are:

(a) New instances of known case frames, scripts, or plans. Each of these can be a kind of novel language in the sense that sentences never seen before can be processed appropriately. This may mean that information is retrieved from a data base on request, or that a representation of a news story is constructed and remembered, or that a question is answered about an earlier dialogue, and so on. If the general rules in a system are good ones, then a relatively small number of rules will allow a program to handle a wide variety of inputs, most of which were never explicitly anticipated by the programmer of the system. This is the simplest type of novel language, and is by now so familiar that it hardly seems to be a way of dealing with novel language at all.

(b) Isolated novel words that have to be understood in context. Some work has been done in this area by Granger[3]. Whenever we can extract a meaning structure for a sentence in context, we have some hope of guessing the meaning of a novel word. For example, if we were told:

When the tank got low, John filled his car with gasohol.

A system that had some scriptal knowledge in the automobile domain could guess that gasohol was a kind of fuel, or possibly a fluid to substitute for oil, water or antifreeze, or by some stretch of the imagination, gasohol might be something to put in a tank that just happens to be being transported by the car. Several types of information can be used to constrain the possible meanings for gasohol: it is something that can be the instrument of "fill", something that a car is filled with, probably its tank, that since the tank got low, something, probably the car or John, was using up the substance in the tank.

(c) Combinations of words that denote items never before known to a system. Examples in (a) above shade into others where concepts are referenced that are novel to a system. For example, complex noun phrases can use familiar words to construct novel items, as in the

†This work was supported in part by the Office of Naval Research under Contract N00014-75-C-0612, and in part by the National Science Foundation under Grants NSF IST 81-17238 and NSF IST 81-20254.

phrase (from Finin[4]):

... engine housing acid damage report summary ...

Here, all the words (engine, housing, etc.) may be known, but the phrase taken as a whole denotes an item that may never have been encountered before by the system. A program that "understands" this phrase could create an internal representation for the item, and infer properties about the item, e.g. that the item was the summary part of a report, that the report was about engine housing acid damage, that the material of the engine housing is probably metal, that the acid damage was to the housing, that acid damage to metal is called "corrosion", and so on. From this information, a system could recognize paraphrases and a variety of references to the same item.

(d) Events that are novel, as in the example:

My dachshund bit our postman on the ear.

Waltz[5] lays out mechanisms that would allow a system to generate the working equivalent of a mental image for this sentence, attempt to simulate the running of a "mental image" corresponding to the sentence, and from the difficulties encountered in running the mental image simulation, judge that the sentence was at least mildly implausible.

(e) New schemas, describing goal-oriented sequences of actions that may never have been encountered before, as in hearing and understanding the nature of skyjacking for the first time[6]. Here, the understanding consists of first untangling the motivations for each of the participants, accounting for each of the actions that are part of the overall schema, and generalizing the schema so that novel occurrences of similar schemas can remind the system of the original schema.

(f) Novel metaphors and analogies. Here the variety of language that requires explanation is staggering. Understanding metaphorical language first requires noting that the language *is* metaphorical, that is that it couldn't be literal descriptive text. (This in turn requires an internal model of what is ordinary, expected, or possible, that a system can use to judge the plausibility of novel language—see item (d) above.) Next, information from the "base domain", that is the domain in which the language has literal meaning, must be somehow transferred (with appropriate modifications) to the "target domain", i.e. the domain which is actually being described. As an example, given the sentence:

John ate up the compliments.

we would want to transfer material such as pleasure, desire and "ingestion" (suitably modified) from the eating domain to the communication domain. The result can become the basis for learning about a new abstract domain or it may simply be that a metaphor allows one to express in a few words many notions about a target domain that would otherwise require a much lengthier exposition. In any case, a system should also keep some record of its metaphor understanding process, so that subsequent processing of similar metaphors would be eased.

In this article, we look in more detail at the problem of designing mechanisms that will allow us to deal with the types of novel language described in (e) and (f) above, namely schema learning and the understanding of metaphors. This work is just beginning. The examples we describe have been chosen to be types that occur commonly, so that rules that we need to understand them can be used to also understand a much wider range of novel language. However, we must note that there is only so far that rules can take us: ultimately the power of systems will depend on the sheer amount of knowledge they have, knowledge which can be used as the base domain for new metaphors, and schemas that can be used to build yet more schemas. Therefore, to really achieve something resembling common sense, we will have to *exercise* our rules on whatever base information we have, building a yet larger base on which the rules can operate recursively. This important process is meant to be a first-order model of the process of adult knowledge acquisition through language.

2. SCHEMA LEARNING

In this section we examine the problem of processing texts that express unfamiliar concepts. Acquiring some grasp of those new concepts is an essential aspect of processing such texts. This is different from learning new words from context. The distinction here is between unfamiliar words that express familiar concepts and familiar words that express unfamiliar concepts. The former problem has been somewhat studied (Selfridge, Granger, Anderson, Langley). The latter has not.

How can familiar words express unfamiliar concepts? After all, knowing a word entails knowing the set of concepts corresponding to its various word senses. While this is true, words in aggregate often can be used to express concepts beyond the simple composition of their meanings. These larger concepts have variously been termed frames[7, 8], schemas[9, 10], scripts[11] or MOPs[12]. Structures corresponding to these larger concepts are used to organize world knowledge in artificial intelligence systems, and play a crucial role in the understanding process in natural language systems (e.g. [13–17]. We will use the (relatively) neutral term "schema" to refer to these knowledge structures.

Very briefly, schemas are used in natural language processing as follows. A text is input to the system. The schemas relevant to the situations described in the text are selected and activated. Schema selection is a difficult problem, outside the domain of this paper. There have been several approaches (e.g. [17–19]).

After schema activation, text sentences are interpreted with respect to the chosen schemas. For each situation the corresponding schema supplies normal causal and temporal connections among events, a specification of what is important and what is not, preconditions and postconditions, etc. Thus, the use of schemas facilitates the task of constructing a unified conceptual representation for the text as a whole. In some systems[17, 20] the schemas are also used to aid in word and sentence interpretation.

Now we can ask a crucial question: What can a natural language system do if it does not have an appropriate schema for understanding a new input text? As a partial answer, we will introduce a new kind of learning called *Explanatory Schema Acquisition*. As the name implies, it is used to acquire schemas. It is not a universal learning technique. The method will be applied only to acquisition of volitional schemas, i.e. schemas used by people in problem solving situations. Furthermore, it builds on knowledge already in the system and so it is not immediately applicable to learning a system's first schemas. Even with non-schema and first schema learning ruled out, a very large and interesting class of learning remains. In fact, it seems that a very large fraction of human adult learning is of this kind. It encompasses learning schemas from instruction, from observation of others, from untutored examples, and from fortuitous accidents.

The main argument that will be advanced is that acquiring schemas involves generalizing structures made up of old and familiar schemas which are combined in novel ways. The generalizing process itself is performed through consideration of the interactions between the effects, preconditions and slot filler constraints supplied by the component schemas.

Thus, the method is a knowledge based one. It is capable of one trial learning. Moreover, it relies very little on inductively acquired correlational experience.

2.1 *An example*

To clarify the procedure, consider an example. This example is a story about a kidnapping, Let us assume that we, the readers of this example, do not yet have a schema for kidnapping or extortion or any similar notion. We do, however, assume the knowledge of a considerable quantity of background information about stealing, bargaining, the use of normal physical objects, and goals of people and institutions.

Example story:

> Paris police disclosed Tuesday that a man who identified himself as Jean Maraneaux abducted the 12-yr-old daughter of wealthy Parisian businessman Michel Boullard late last week. Boullard received a letter containing a snapshot of the kidnapped girl. The next day he received a telegram demanding that 1 million francs be left in a lobby waste basket of

the crowded Pompidou Center in exchange for the girl. Asking that the police not intervene, Boullard arranged for the delivery of the money. His daughter was found wandering blindfolded with her hands bound near his downtown office on Monday.

A KIDNAPPING schema, if we had one, would contain information to help us make sense of the story. With it, processing the story would be relatively easy.

But by assumption we do not know about kidnapping. Therefore some events in the story are incomprehensible. In particular we cannot explain why Maraneaux might steal Boullard's daughter. While this is quite clearly an instance of taking something that belongs to someone else, there is no motivation for it. The daughter has no apparent value to Boullard; a person, unlike money, cannot be used to acquire other valued goods. Any schema-based understander requires motivations for major volitional actions (such as a character invoking the STEAL schema). Therefore, this input seems anomalous.

The confusion is resolved by the next sentence. This input invokes the BARGAIN schema. We know immediately the motivation for Maraneaux trying to bargain with Boullard: he is trying to acquire money. Possessing money is a common goal that can be attributed to most people. Thus, it serves as an understandable motivation for the bargaining. Furthermore, stealing the girl is now motivated: Maraneaux used the STEAL schema to satisfy a precondition of the BARGAIN schema. The precondition states that the bargain is unlikely to work unless each party indeed possesses the item he plans to trade away.

Thus far we have done nothing new. Previous systems have proposed understanding new text inputs via analysis of goals and plans of the characters[2, 16, 21]. These systems tend to be more oriented toward "planning" or "problem solving" than "script application".

Once the story has been understood in this way it might already be viewed as a new schema. The system could file away the representation as a method by which a particular person (Maraneaux) can procure a particular amount of money (1 million francs) by a particular action (stealing Boullard's daughter and offering to trade her back for the money). This is a mistake for several reasons. The most important is that it is simply far too specific.

Our concern here is how a system might do better than to simply file away a very specific plan. Our contention is that the same knowledge used to process the input in the first place can be used to make the schema more general. For example, the system has the knowledge necessary to prove that if Maraneaux wanted 100,000 francs instead of 1 million, that the same plan would work. It can do this because the system knows the function of the million francs in Maraneaux's plan. It knows that the money is traded by Boullard for the return of his daughter. Also it knows that the preconditions for Boullard's acceptance of the proposed bargain are that (1) Boullard must value his daughter's safety more than the money and (2) that Boullard must have access to that amount of money. Clearly, since 1 million francs satisfies these requirements, any amount less than 1 million francs also satisfies the requirements and would have worked. Sums larger than a million francs might work as well provided they do not violate (1) or (2) above. We have been a bit sloppy in our analysis. To understand Maraneaux's actions it is not important in reality for Boullard to have access to the money but only for Maraneaux to *believe* he does, and for Maraneaux to *believe* Boullard values his daughter. Nonetheless, the point is well made: this event can be generalized through knowledge-based manipulations using information that had to be in the system anyway in order for the story to be understood. In a like manner the identity of Boullard, his daughter, and Maraneaux are not important. What is important are that these roles be played by people with certain relationships to other people and things. The required relationships are dictated by the volitional actions required of the people by the schema. After these knowledge-based generalizations have been made, the specific event can be transformed into a KIDNAP schema.

In general, the newly generalized schemas require further refinement. Due to eccentricities in the input story, the schema may lack information. For example, if the first kidnapping story seen by the system reported the kidnappers successfully escaping with the ransom even though they killed the hostage, the system might acquire a distorted concept of kidnapping. Even more frequent are cases where the first schema constructed is correct but incomplete. This might result from situations where there are alternate methods of achieving certain sub-goals, only one of which is reported. Clearly, schema modification is essential. Thus, the system's schemas must constantly be adjusted and refined in reaction to normal input processing.

2.2 *The generalization process*

There are two problems that the generalization process must face. The first is to know when it should be applied. Clearly, every input text ought not to cause the system to construct a new schema. Only "interesting" inputs should invoke the schema acquisition system. The second problem is how to perform the generalization. There are a number of subproblems here, for example, selecting which events and objects should be generalized, imposing limits on the extent of generalization, and actually carrying out the schema modification.

There are four situations which when recognized in the text either individually or in combination ought to invoke the generalization routines. They are:

Schema composition
Secondary effect elevation
Schema alteration
Volitionalization.

In the first part of this section we will illustrate each of these situations with an example.

2.2.1 *Schema composition.* The first situation we will discuss is called *schema composition.* Basically, it involves composing known schemas in a novel way. Typically, this will involve a primary schema, essentially unchanged, with one or more of its preconditions satisfied in a novel way by other known schemas.

An example of this was seen in the above kidnapping story. In that story, the primary schema is BARGAIN, a schema which we assumed the system already knew. One of the preconditions specified in the BARGAIN schema is that each party to the bargain must convince the other that he can indeed deliver his side of the bargain. For Maraneaux, this corresponds to making Boullard believe that he (Maraneaux) has control of Boullard's daughter an can, therefore, relinquish the girl to him. Maraneaux achieves this by actually establishing control over the daughter (via an instance of the STEAL schema) and then sending Boullard a photograph. To the system, this is a novel way to satisfy BARGAIN's preconditions. We know this must be novel to the system because if it were not, the system would already have a schema in which this precondition of BARGAIN was satisfied by an application of STEAL. But by hypothesis, the system does not yet possess a kidnapping schema and therefore, cannot yet know of this method of satisfying the precondition. Thus, a precondition of a known schema has been satisfied in an interesting new way, and a new schema must be constructed to capture the underlying generalization.

2.2.2 *Secondary effect elevation.* Consider the following scenario:

> Fred wanted to date only Sue, but Sue steadfastly refused his overtures. Fred was on the verge of giving up when he saw what happened to his friend, John: John wanted to date Mary but she also refused. John started seeing Wilma. Mary became jealous and the next time he asked her, Mary eagerly accepted. Fred told Sue that he was going to make a date with Lisa.

Here Fred has not acquired a new schema; he has used an existing schema (DATE) in a new way. This is called secondary effect elevation. Fred's DATE schema already contains all of the knowledge necessary for resolving his dilemma. The problem is that the normal DATE schema is organized in the wrong way. In secondary effect elevation situations an existing schema is annotated indicating that the schema may be used to achieve a result which is normally neutral or negative.

The main purpose of the DATE schema is to satisfy certain recurring social goals (like companionship, sex, etc.). DATE contains secondary effects as well. These are often undesirable effects accompanying the main, planned effects. For example, one is usually monetarily poorer after a date. Another secondary effect is that if one has an old girlfriend, she may become jealous of a new date.

What Fred learned from John's experience is that it is occasionally useful to invoke the DATE schema in order to cause one of its secondary effects (jealousy) while completely ignoring the usual main goal.

Just as with schema composition, the existing schema is changed to reflect a *generalization* made from a specific instance. In this case, the specific instance is John's interactions with Mary. Notice, however, that Fred did not simply copy John's actions. John actually made a date with Wilma while Fred only expressed an intention to date Lisa. This is not an earth-shaking difference, but in the context of dating it is extremely significant. In the normal DATE situation expressing an intention to date someone is not nearly so satisfying as an actual date. Once modified for the purpose of causing jealousy, however, expressing an intention for a date and actually carrying it out can be equally effective.

One might argue that the distinction between main and secondary effects of a schema is otiose and, in situations such as this, even deleterious. After all, DATE already had all of the information necessary for solving Fred's problem. If a system simply treats all of the effects of a schema the same, then any effect can be singled out during the planning process to be used as the main goal. There is, however, a strong argument against this position. The possible desired effects of a schema do not exist only within the schema itself. They are used to organize and select among schemas in both understanding and planning applications (see [14] and [18]). Many effects (like feeling more tired after a date than before) will not be used in the normal planning or understanding process. If they are treated the same as legitimate main goals the system will be swamped in a combinatorial quagmire of undifferentiated possibilities, most of which are wildly implausible. For example, we do not want our understanding process to predict that John will take a nap when it is told that John dated Mary. Given the input "John took a nap" the system ought to be able to justify it. However, it ought not actively predict it. Given the multiplicity of individual actions making up the DATE schema (each with its own set of effects) the vast majority of the effects from this scheme (and any other schema) are simply irrelevant to overall planning and understanding processes. Instead, we would like our system to single out the plausible volitional effects of its schemas and use only those for schema organization and selection. Thus, in our example, Fred has constructed, via secondary effect elevation, a new use of the DATE schema.

2.2.3 *Schema alteration.* Schema alteration involves modifying a nearly correct schema so that it fits the requirements of a new situation. The alteration process is guided by the system's world model. This is illustrated by the following brief anecdote:

> Recently I had occasion to replace temporarily a broken window in my back door with a plywood panel. The plywood sheet from which the panel was to be cut had a "good" side and a "bad" side (as does most raw lumber). The good side was reasonably smooth while the bad side had several ruts and knot holes. I automatically examined both sides of the sheet (presumably as part of my SAWING or CUTTING-A-BOARD-TO-FIT schema) and selected the good side to face into the house with the bad side to be exposed to the elements. After I had cut the panel and fitted it in place I noticed that several splinters had been torn out leaving ruts in the "good" side. I immediately saw the problem. Hand saws only cut in one direction. With hand saws, the downward motion does the cutting while the upward motion only repositions the cutting blade for another downward motion. I had cut the wood panel with the "good" side facing down. The downward cutting action has a tendency to tear splinters of wood out of the lower surface of the board. Since the good side was the lower surface, it suffered the loss of splinters. If I had to perform the same action again, I would not make the same mistake. I would cut the board with the good side facing up. However, what I learned was not just a simple specialized patch to handle this particular instance of splintering. Since I knew the cause of the splintering, I knew that it would not always be a problem: it is only a problem when (1) the lumber is prone to splintering, (2) there is a "good" side of the board that is to be preserved, and (3) one is making a crosscut (across the wood's grain) rather than a rip cut (along the grain). Moreover, the solution is not always to position the wood with the good side up. My electric saber saw (also a reciprocating saw) cuts during the upward blade motion rather than the downward motion. Clearly, the solution when using the saber saw is the opposite: to position the board with the good side down. Now, these are not hard and fast rules: with a sufficiently poor quality sheet of plywood splintering would likely always be a problem. Rather, these are useful heuristics that lead to a refinement of the SAWING schema.

Note that this refinement to the SAWING schema is far more general than required to handle the particular problem that gave rise to it. The refinement contains contingencies relevant to the

use of saber saws even though no saber saw was used in the immediate problem. This is possible because the refinement is driven by world model, not just the problem. The SAWING schema was altered by identifying and eliminating the offending cause in the underlying knowledge-based explanation of the phenomena.

2.2.4 *Volitionalization.* This situation involves transforming a schema for which there is no planner (like VEHICLE-ACCIDENT, ROULETTE, etc.) into a schema which can be used be a planner to attain a specific goal. Consider the following story:

> Herman was his grandfather's only living relative. When Herman's business was failing he decided to ask his grandfather for a loan. They had never been close but his grandfather was a rich man and Herman knew he could spare the money. When his grandfather refused, Herman decided he would do the old fellow in. He gave him a vintage bottle of wine spiked with arsenic. His grandfather died. Herman inherited several million dollars and lived happily ever after.

This story is a paraphrase of innumerable mystery stories and illustrates a schema familiar to all who-done-it readers. It might be called the HEIR-ELIMINATES-BENEFACTOR schema. It is produced via volitionalization by modifying the existing non-volitional schema INHERIT. INHERIT is non-volitional since there is no active agent. The schema simply dictates what happens to a person's possessions when he dies.

In this example, volitionalization parallels schema composition. One of the preconditions to INHERIT is that the individual be dead. The ELIMINATE-BENEFACTOR schema uses the schema MURDER to accomplish this. One major difference is that schema composition requires all volitional schemas. This parallelism need not always be present, however. Non-volitional to volitional transformation is also applicable to removing stochastic causal steps from a schema resulting in a volitional one.

2.3 *Limits on generalization*

Basically, the generalization process is based on certain data dependency links established during understanding.

After a story is understood, the understood representation can be viewed as an *explanation* of why the events are plausible. For example, take the case of a kidnapping. KIDNAP is an instance of schema composition, not unlike RANSOM. Thus, the first kidnapping story seen by the system is understood as a THEFT followed by a BARGAIN. If the kidnapper is successful, the ransom is paid. For a system to understand this, it must justify that the person paying values the safety of the kidnapped victim more than the ransom money. This justification is a data dependency[22] link to some general world knowledge (e.g. that a parent loves his children). Now the event can be generalized so long as these data dependency links are preserved. Clearly, as long as the data dependencies are preserved, the underlying events will still form a believable whole.

Consider again the secondary effect elevation example of Fred trying to date Sue. The observed specific instance is John's interactions with Mary. Notice, however, that Fred did not simply copy John's actions. John actually made a date with Wilma while Fred only expressed an intention to date Lisa. This is not an earth-shaking difference, but in the context of dating it is extremely significant. In the normal DATE situation expressing an intention to date someone is not nearly so satisfying as an actual date. Once modified for the purpose of causing jealousy, however, expressing an intention for a date and actually carrying it out can be equally effective. That is, they both maintain the data dependency link for why we believe that Sue is in fact jealous.

Likewise, in the alteration example the schema for preserving one side of a board while sawing can be generalized. The resulting schema is applicable to circular saws, jig saws, etc. as well as hand saws. Again this is due to the preservation of a data dependency link: we believe that the wood's surface is preserved because the surface is supported by the rest of the board during deformation due to the saw's teeth. As long as we know which direction the teeth point on a saw, we know how to orient the board to preserve its good side.

2.4 *Comparison to previous work*

How does this method compare to other learning systems? There are a number of previous learning systems that spring to mind: Schank's MOPs[12], Selfridge's language learning model[36],

Soloway's program to learn the rules of baseball[25] and SRI STRIPS system[37]. The system outlined is strikingly different from Schank's and Selfridge's. It has some interesting similarities to Soloway's and one part of the STRIPS system.

While the domain of Schank's MOPs is similar to the described system, the learning technique used with MOPs is very different. The systems of Kolodner[38] and Lebowitz[20] both made "generalizations" but these are all of the correlational variety and might better be termed "specializations". IPPS generalization that Italian terrorists tend to shoot people in the knee caps, for example, is actually a correlational constraint noticed in the pre-existing terrorism MOP. The result is actually a specialized terrorism MOP to be applied only to Italian terrorist stories which makes a prediction about shooting in knee caps. Learning in both IPP and CYRUS is of this variety. Their approach precludes the kind of learning that extends a system's range of processing. Lebowitz's general terrorism MOP could not in principle be learned by his system. In the example outlined, the system learned an EXTORT schema without having a more general version already built in.

Selfridge's system was concerned with learning sentence structure and the names of already existing concepts. It learned, for example, that the words "put on" can refer to the already defined algorithmic concept "get dressed in". The domain of my system is learning the original concepts. It might be interesting to explore how these ideas could be applied to language learning but that would not be the main thrust.

Soloway's system is similar to the one outlined here in that it has the flavor of one-trial or "insight" learning. Furthermore, he made use of general background goal information (in the form of notions such as competition) to aid in processing. However, the domain of learning baseball rules from game descriptions is very different from learning process schemata. Also, the purpose of his system is very different. It did not try to extend the range of its processing in an open-ended way. Rather, it tried to induce general rules from instances. In that sense it is more of an inductive inference system.

The MACROPS idea of SRIs are similar in that they result in new processing structures which can in turn be combined to form yet other structures. However, the domain of planning paths around blocks and through doors is much more constrained and simplified. Furthermore, the MACROPS structures were built from a successful planning search through the problem space, not in the midst of processing inputs. This makes STRIPS very inward motivated in its learning.

2.5 *Conclusion*

There are several concluding points:

(1) Explanatory schema acquisition does not depend on correlational evidence. Unlike some learning system (e.g. [23, 24]), it is capable of one trial learning. It is somewhat similar to Soloway's view of learning[25].

(2) The approach is heavily knowledge-based. A great deal of background knowledge must be present for learning to take place. In this respect explanatory schema acquisition follows the current trend in AI learning and discovery systems perhaps traceable to Lenat[26].

(3) The learning mechanism is not "failure-driven" as is the MOPs approach[12]. In that view learning takes place in response to incorrect predictions by the system. In explanatory acquisition learning can also be stimulated by positive inputs which encounter no particular problems or prediction failures.

(4) The absolute representation power of the system is not enhanced by learning new schemas. This statement is only superficially surprising. Indeed, Fodor[27] implies that this must be true of all self-consistent learning systems. Explanatory schema acquisition does, however, increase processing efficiency. Since all real-world systems are resource limited, this learning technique does, in fact, increase the system's processing power. Furthermore, it may indicate how Socratic method learning is possible and why the psychological phenomenon of functional fixedness is adaptive.

3. UNDERSTANDING METAPHOR

3.1 *Importance of metaphor*

Metaphors are pervasive. It is nearly impossible to avoid metaphor in language use, even if

the language is technical. For example, hydraulic metaphors are common in economics (e.g. economic *pressure*, cash *flow*, *turning off* the money supply, *draining* of assets, etc.). It is not possible to talk about *love* except through metaphor: love can be likened to a journey together, a meeting of minds, complementary shapes (as in fitting or belonging together), madness, falling into an abyss, transmitting and receiving on the same wavelength, and so on. Jackendoff [28] has argued that metaphor is the basic process by which we acquire proficiency in abstract domains; he suggests that as infants, when we encounter a novel domain, we use existing sensory-motor schemas to form the basis of schemas suitable for understanding the abstract domain, and that this process can continue recursively, using existing abstract schemas as the basis for understanding novel abstract domains. Jackendoff therefore suggests that the surface similarity of "Mary kept the ring in a box" and "They kept the business in the family" reflects a deep similarity due to the derivation of the abstract domain of *possession* from the concrete domain of *position.*

Metaphors can be used to transfer complex combinations of information from one well-known domain to another less well known or completely unfamiliar one. Understanding metaphorical language first requires noting that the language *is* metaphorical, that is that it couldn't be literal descriptive text. This in turn requires an internal model of what is ordinary, expected, or possible, that a system can use to judge the plausibility of novel language (see for example item (d) in the Introduction). Next, material from the "base domain", that is the domain in which the language has literal meaning, must be used to understand the "target domain", that is, the domain which is actually being described. This could be done in a number of ways, for example, by establishing links between the base domain of the metaphor and the target (novel) domain that the metaphor is being used to describe, or by copying base domain structures into a target domain. The result can become the basis for learning about a new domain (by transferring knowledge from the base domain selectively) or it may simply be that a metaphor allows one to express in a few words many notions about a target domain that would otherwise require a much lengthier exposition. Consider for example:

(S1) John ate up the compliments.

or

(S2) Robbie's metal legs ate up the space between him and Susie.†

Assuming that these sentences represented novel uses of the words "ate up", we might want a system to infer that in the first sentence John desired the compliments, eagerly "ingested" them with his mind, thereby making them internal and being given pleasure by them, and that in the second sentence, the distance between Robbie and Susie was being reduced to zero, just as an amount of food is reduced to zero when it is "eaten up".

In the following sections I will show methods which will make the correct interpretations of the two examples above. First, however, I must introduce "event shape diagrams", a new representation scheme for verb meaning, which is used centrally in this method for understanding novel metaphors.‡

3.2 *Event shape diagrams*

In their simplest forms, event shape diagrams have a time line, a scale, and values on the scale at one or more points. Diagrams can be used to represent concurrent processes, causation, and other temporal relations by aligning two or more diagrams, as illustrated in Fig. 1 which shows the representation for "eat". Note that several simple diagrams are aligned, and that each has different kinds of scales, and different event shapes. The top scale corresponds to the CD primitive INGEST [29]. Causal relations hold between the events described in each simple

†This is a slightly modified sentence from Isaac Asimov's *I, Robot.*

‡Only verb-based metaphors will be treated here. These methods seem inappropriate for interpreting noun-based metaphors such as "John is a rat", or for "phenomenological metaphors", such as "I woke up in the morning with a sledge hammer banging in my head", as well as for others, no doubt. I have not attempted a taxonomy of metaphor types.

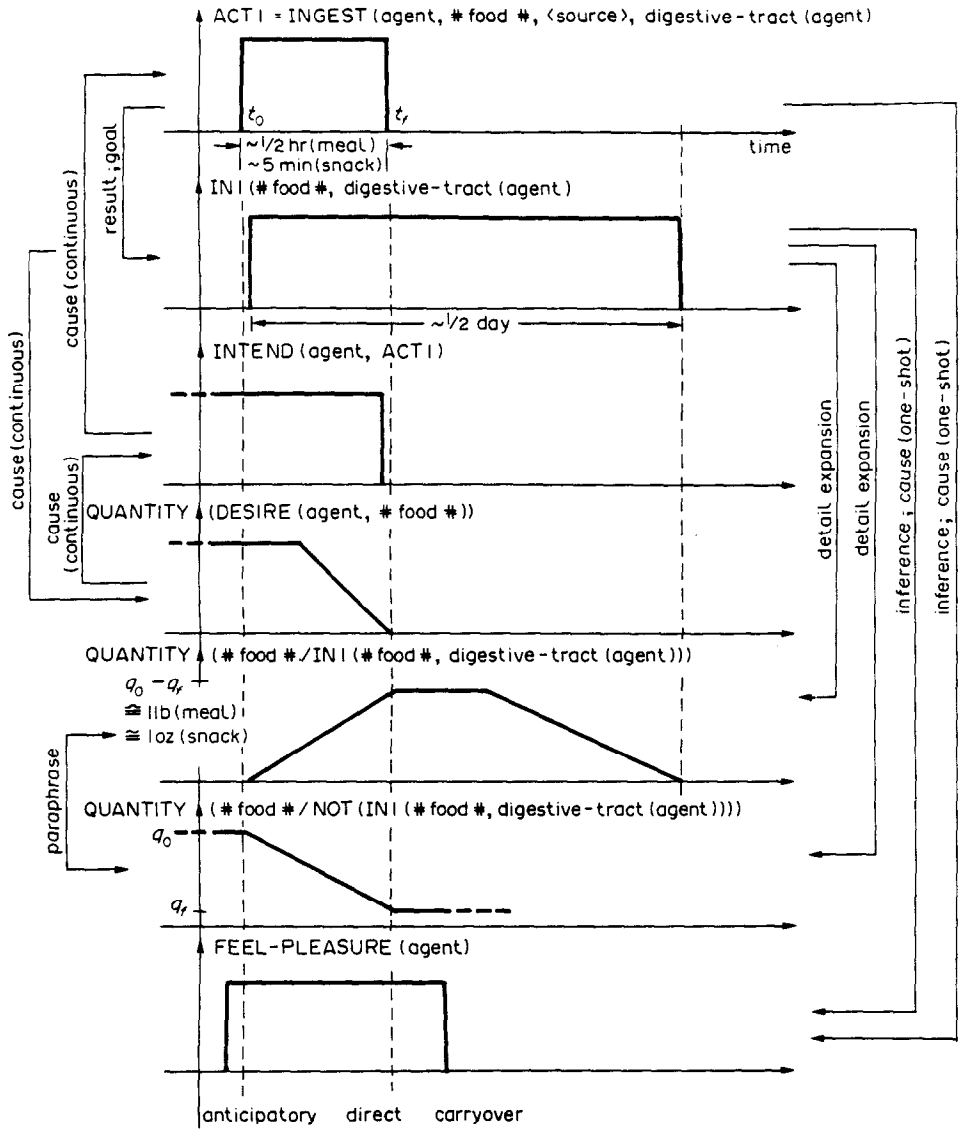

Fig. 1. Event shape diagram for "eat".

diagram. The names for the causal relations are adopted from Rieger's CSA work[30]. The action INGEST stops in this default case where "desire to eat" goes to zero. "Desire to eat" sums up in one measure coercion, habit, and other factors as well as hunger. Typical values for amounts of food, time required to eat, and so on are also associated with the diagram, to be used as default values.

Many adverbial modifiers can be represented neatly: "eat quickly" shrinks the value of $t_f - t_0$ with respect to typical values; "eat a lot" increases the values of $q_0 - q_f$ above typical values. Similarly "eat only half of one's meal", "eat very slowly", "eat one bite", etc. can be neatly represented. "Eat up" can be represented by making the

QUANTITY(food/IN 1 (food, digestive-tract (agent)))

go to zero before the DESIRE (agent, ACT 1) goes to zero. This representation is shown in Fig. 2.

The point of time from which events are viewed can also be clearly represented. Past tense (e.g. "we ate 3 hamburgers") puts "now" on the time line to the right of the action, while future tense puts "now" to the left of the action, and present progressive (e.g. "we are eating") puts "now" between t_0 and t_f.

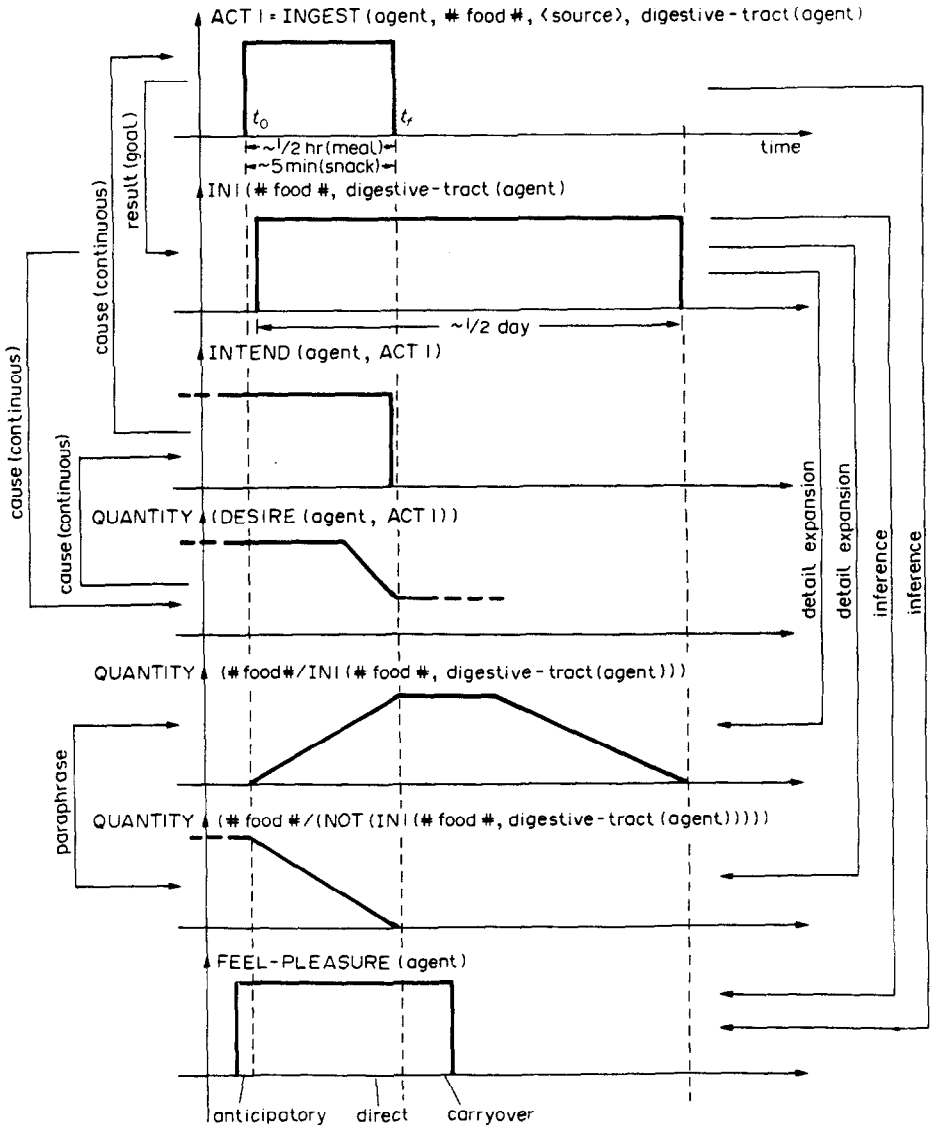

Fig. 2. Event shape diagram for "eat up".

More levels of detail can be added if needed. For instance, the action diagram for eating ought to have links to more general event shape diagrams representing the typical daily eating habits of humans (three meals, one in the early morning, one around noon, and one in the early evening, plus between-meal snacks, coupled with diagrams representing the gradual onset of desire to eat after a meal); the diagram for "eating" should also have links to more detailed event shape diagrams that expand upon the actions involved (eating involves many recurrences of putting food in one's mouth, biting, chewing and swallowing, and the diagram for the amount of food inside the agent can reflect a series of stepwise changes as each mouthful is ingested).

For more detail on event shape diagrams, see [31].

3.3 *Metaphor with event shape diagrams*

The interpretation of verb-based metaphors is based on the following general principles:

(1) Both verbs and nouns have inherent selection restrictions. Thus, for the purposes of this example, "eat (up)" prefers that its semantic object be food, and foods of various kinds are marked by a preference to appear with certain actions, such as "eat", "buy", "grow", "prepare", "throw away", etc. (See Finin[4] for discussion of "case frames" for nouns.)

(2) Nouns are far less likely to be metaphorical than verbs. If a verb and object do not

match each others' selection restrictions, the object should be taken as referring literally, and the verb as referring metaphorically. Thus, we can correctly predict that each of the following sentences is really about ordinary actions on food, even though literally these actions are very remote meanings for each of the verbs:

(S3) Mary destroyed the food. (= prepared badly or ate ravenously).
(S4) Sue made the food disappear. (= ate up rapidly).
(S5) John threw the food together. (= prepared rapidly).

(3) Understanding of a verb-based metaphor involves (a) selection of candidate meanings using the semantic object, (b) matching the event shape diagrams of the candidate meanings with both the current context and the event shape diagrams of the *actual* verb in the sentence.

If there is more than one basic meaning candidate for a metaphorically used work (as in S2 above) the most appropriate meaning is selected by testing the various basic meanings in the current context to see which fits best. Once a basic meaning is selected, the event shape diagrams of this meaning are matched with the event shape diagrams of the *actual* verb used, and some meaning is transferred. The meaning transfer can take two forms: (1) modifying the basic meaning, in a manner similar to adverbial modification; and (2) (more interestingly) superimposing certain portions of the event shape diagram for the verb actually used in the sentence onto the selected basic meaning.

This process should be clearer after I show examples of its operation on sentences (S1) and (S2).

3.4 *An example*

Consider the processing required to handle the metaphor in

(S1) John ate up the compliments.

Using principle (1) above, we first note that "ate up" prefers *food* of some kind as a semantic object, that "compliments" is not a food, and itself prefers an MTRANS-type verb[24], in particular either "tell" or "hear". Next, using principle (2), we can judge that "compliments" refers literally, and so either "tell" or "hear" is probably the true basic verb. The event shape diagrams for "tell" and "hear" are shown in Fig. 3. STM means "short term memory" and LTM means "long term memory". These terms are used here with their common sense (non-technical) meaning.

If the sentence appeared in context, we might be able to select the proper basic meaning by comparing the two possibilities with our current expectations, but in this case, we have to rely on event shape diagram matching to determine the best choice.

Let us look first at trying to match "tell" with "eat up". In order to judge the quality of the match, we must first describe a scoring scheme. The scoring scheme used here is rather simple: it looks for scales that are the same, and matches them, provided the shapes of the scale are the same (i.e. both are changes in the positive direction, or both are *occurrences*, where an occurrence is defined as a change on some scale from a zero to a non-zero value, followed by a change back to zero again. In this case, MTRANS matches INGEST—both are *occurrences*—*and*

INTEND (agent, MTRANS (agent, compliment, STM (agent), STM (hearer)))

matches

INTEND (agent, INGEST (agent, food, [source], digestive-tract (agent)))

—both are *negative changes*. There is a serious mismatch between these two, in that STM (hearer) does not match digestive-tract (agent) well, and these items are the goal portions of the DESIRE, the most important part.

Now consider the match between "hear" and "eat up". As before, MTRANS matches

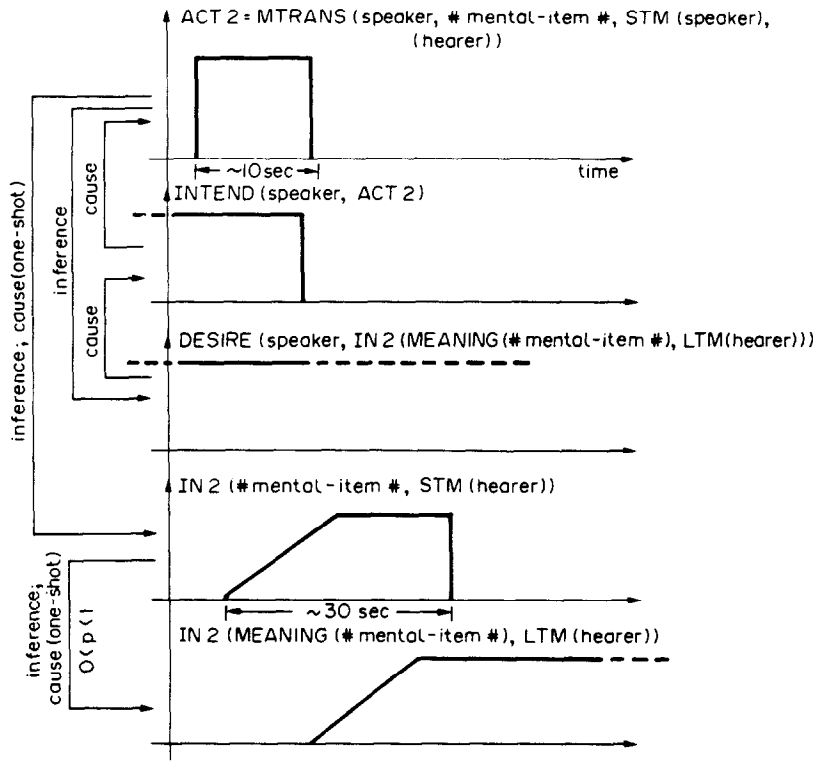

Fig. 3(a). Event shape diagram for "tell".

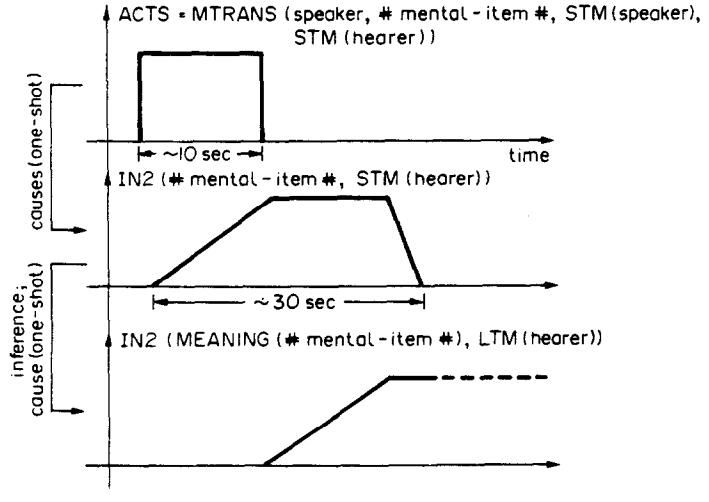

Fig. 3(b). Event shape diagram for "hear".

INGEST, but now the INTEND postion of "eat up" has no match. However, IN1 (compliment, STM (hearer)) matches IN2 (food, digestive-tract (agent)) very well—both are the major scales of their respective verbs, and both have the same "shape", namely the *occurrence* shape, and finally, IN1 and IN2 are closely related binary predicates.

The understanding of the metaphor can now be addressed. Understanding in this model is the transfer to *hear* of the "residue" of the meaning of *eat up*, where by "residue" I mean the portion of *eat up* that had no match with portions of *hear*. The residue in this case consists of the scales for DESIRE, INTEND, QUANTITY and FEEL-PLEASURE that were associated with *eat up*. Theoretically, there are two main options for the mechanism that makes the

transfer: (1) the scales may simply be added to the meaning of *hear*, or (2) some of these scales may already be present in latent or potential form as part of our understanding of *hear*, and the transfer would then consist of boosting their prominence, assigning a polarity to them, etc. Even within this single example, there are three kinds of issues that lead me to believe that option (2) is the right choice in general: first, it is difficult to understand why INTEND cannot be transferred to *hear* unless one realizes that hearing a particular item is not something we can ever intend in a causal sense; second, the transfer cannot be literal in any event—for example we would not want to infer that compliments remain in our STM for a day, just because food may do so; and third, adverbial modification seems to already require scales to be present in latent form, as for example in

(S6) I heard the compliments with great pleasure.

Taking the second option, then, we can construct a meaning for (S1), as shown in Fig. 4. Figure 4(a) shows the enriched version of *hear* used to receive the transferred material from *eat up*. Note that although the items below the dotted line are truly part of the meaning of *hear*, these items would not ordinarily be evoked when understanding the word *hear*, and that really, this version of *hear* represents three meanings, corresponding to "hear", "hear with pleasure",

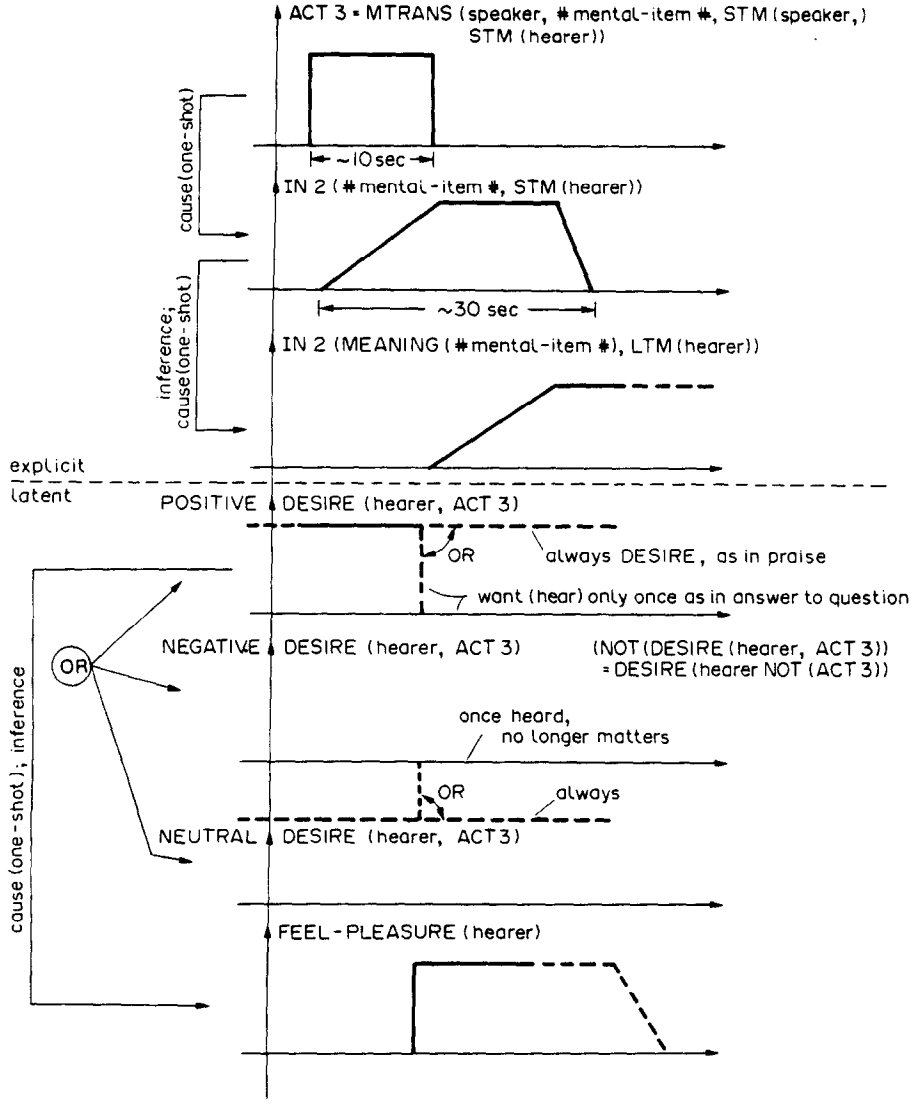

Fig. 4(a). "encircled" event shape diagram for "hear".

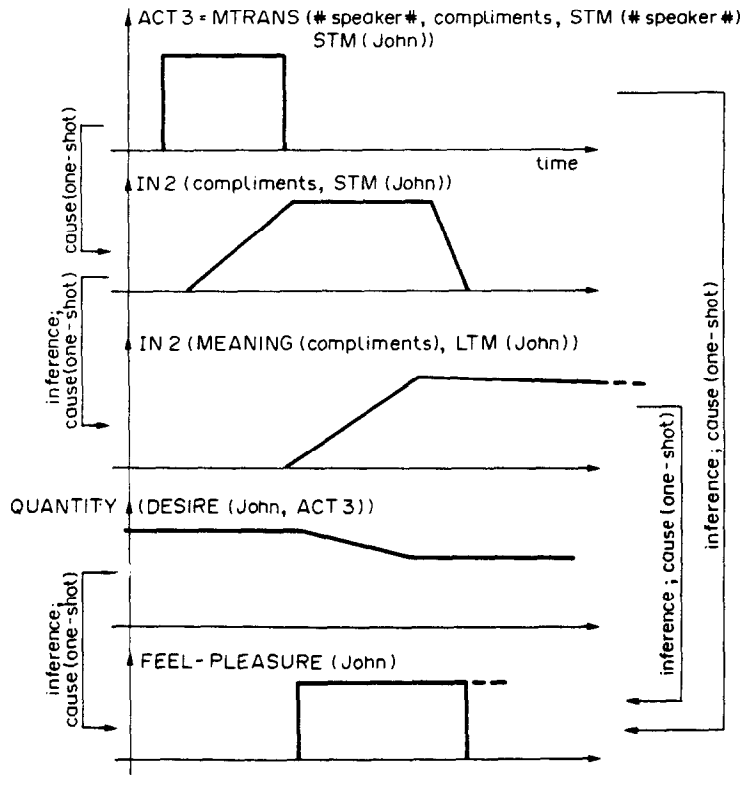

Fig. 4(b). Representation of (S1) "John ate up the compliments".

and "hear with displeasure". It would clearly not be difficult to select "hear with pleasure" by matching with "eat up". Figure 4(b) shows the final meaning representation for (S1).

Example (S2)

(S2) Robbie's metal legs ate up the space between him and Susie.

can be understood using similar methods, though there are some interesting differences. The object of the verb in this case is "space" which is again not an appropriate object for use with "eat up". Again taking the semantic object as the item most likely to refer literally, space suggests that the true basic verb in the sentence ought to be PTRANS, i.e. the physical transfer of an object through space. "Legs" also play an important part here, constraining the PTRANS to be either "run" or "walk" (this requires different processing methods that I have not yet investigated very thoroughly). For our purposes, "run" and "walk" look pretty much the same. There are some main variants that I believe ought to be represented differently, namely the meaning suggested by phrases such as run from (away from) *x*, run to (toward) *y*, run (without source or goal), run from *x* to *y*, and so on. These differ according to whether movement is stated with reference to a source, goal, neither or both, and whether or not the motion actually starts and/or ends at the source and goal points, or whether these specify only the direction of motion. In this case, the QUANTITY of food which goes to zero should make it possible to match the "run to" meaning.

So far, so good, but some interesting issues remain. First, there is little residue to transfer in this case, except for the intensification of the DESIRE to be at the goal. In fact, I don't think that this is bad, but there are some inferences that I make in hearing (S2) that cannot be easily accounted for using this model. In particular, there is an analogy between taking bites and taking steps, and perhaps more important (and possibly related) (S2) seems to focus on the past progressive aspects of the action; to my mind the sentence is better paraphrased as "Robbie was running toward Susie" than as "Robbie ran to Susie". Overall, however, the account of the understanding of the two metaphors seems to capture roughly the right meanings in a natural and (to me) quite satisfying manner; the problems seem to require refinements to the method rather than complete rethinking.

3.5 *Assessment*

I do not want to claim that all metaphors can be handled by methods of the sort that have been described above. I do believe that the mechanisms suggested above are particularly good and natural for a reasonably rich class of metaphors. There still are holes in the theory, however. Consider the following sentence (due to Gentner[32]):

(S7) The flower kissed the rock.

I have suggested that objects ought to be taken literally, and indeed, if we do so, we can obtain a reasonable reading, namely that a flower bent over and its "face" touched a rock gently. However, one could also take the verb literally, and take "rock" and "flower" metaphorically; in this case, the sentence could refer to a gentle woman literally kissing a tough man.

4. CONCLUSION

This work is just beginning. The examples we describe have been chosen to be types that commonly occur, so that rules needed to understand them can also be used to understand a much wider range of novel language. However, we must note that there is only so far that rules can take us: ultimately the power of systems will depend on the sheer amount of knowledge they have, knowledge which can be used as the base domain for new metaphors, and schemas that can be used to build yet more schemas. Therefore, to really achieve something resembling common sense, we will have to *exercise* our rules on whatever base of information we have, building a yet larger base on which the rules can operate recursively.

REFERENCES

1. Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine. *Comm. ACM* **10**(8), 474–480 (1966).
2. K. M. Colby, B. Faught and R. Parkinson, Pattern matching rules of the recognition of natural language dialogue expressions. Stanford A.I. Lab., *Memo AIM*-234 (1976).
3. R. H. Granger, FOUL-UP: a program that figures out meanings of words from context. *Proc. IJCAI*-77, M.I.T., Cambridge Mass., pp. 172–178, August 1977.
4. T. W. Finin, The semantic interpretation of nominal compounds. *Tech. Rep. T*-96, Coordinated Science Lab., Univ. of Illinois, Urbana, March 1980.
5. D. L. Waltz, Toward a detailed model of processing for language describing the physical world. In *Proc. IJAI*-81, Vancouver, B.C., Canada, pp. 1–6, August 1981.
6. G. DeJong, Automatic schema acquisition in a natural language environment. In *Proc. 2nd Annual Nat. Conf. Artificial Intell.*, Pittsburgh, Penn., August 1982.
7. M. Minsky, A framework for the representation of knowledge. M.I.T. AI. *Rep. TR*-306, M.I.T., Cambridge, Mass. (1974).
8. E. Charniak, A framed PAINTING: the representation of a common sense knowledge fragment. *Cognitive Sci.* **4**, 355–394 (1976).
9. D. Bobrow and D. Norman, Some principles of memory schemata. In *Representation and Understanding* (Edited by D. Bobrow and A. Collins). Academic Press, New York (1975).
10. W. Chafe, Some thoughts on schemata. In *Proc. Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Mass., June, 1975.
11. R. Schank and R. Abelson, *Scripts Plans Goals and Understanding.* Erlbaum, Hillsdale, New Jersey (1977).
12. R. Schank, Language and memory. *Cognitive Sci.* **4**, 243–283 (1980).
13. R. Cullingford, Script application: computer understanding of newspaper stories. *Res. Rep.* 116, Yale Computer Science Department, New Haven, Conn., (1978).
14. E. Charniak, MS. MALAPROP, a language comprehension system. In *Proc. 5th IHCAI*, Cambridge, Mass. (1977).
15. D. Bobrow, R. Kaplan, M. Kay, D. Norman, H. Thompson and T. Winograd, GUS, a frame driven dialog system. *Artificial Intell.* **8**(1) (1977).
16. R. Wilensky, Understanding goal-base stories. Ph.D. dissertation, Yale Computer Science Rep. 140, Yale University, New Haven, Conn. (1978).
17. G. DeJong, Skimming stories in real time: an experiment in integrated understanding. *Res. Rep.* 158, Yale Computer Science Department, New Haven, Conn. (1979).
18. E. Charniak, With spoon in hand this must be the eating frame. In *Proc. 2nd Workshop on Theoretical Issues in Natural Language Processing*, University of Illinois, Urbana, Ill. (1978).
19. S. Fahlman, *NETL: A System for Representing and Using Real-World Knowledge.* M.I.T. Press, Cambridge, Mass. (1979).
20. M. Lebowitz, Generalization and memory in an integrated understanding system. Computer Science, *Res. Rep.* 186, Ph.D. dissertation, Yale University, New Haven, Conn. (1980).
21. C. Schmidt and N. Sridharan, Plan recognition using a hypothesize and revise paradigm: an example. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pp. 480–486, 1977.

22. J. Doyle, Truth maintenance systems for problem solving. M.I.T. A.I. *Tech. Rep. TR*-419, M.I.T., Cambridge, Mass. (1978).
23. P. H. Winston, Learning structural descriptions from examples. *Rep. AI TR*-231, M.I.T. A.I. Lab., Cambridge, Mass.
24. M. Fox and R. Reddy, Knowledge-guided learning of structural descriptions. In *Proc. 5th IJCAI*, Cambridge, Mass., 1977.
25. E. Soloway, Learning = interpretation + generalization: a case study in knowledge-driven learning. *COINS Tech. Rep.* 78-13, University of Massachusetts, Amherst, Mass. (1978).
26. D. B. Lenat, AM: an artificial intelligence approach to discovery in mathematics as heuristic search. *SAIL-AIM*-286, Standord University (1976).
27. J. Fodor, *The Language of Thought*. Crowell, New York (1975).
28. R. Jackendoff, A system of semantic primitives. In *Theoretical Issues in Natural Language Processing* (Edited by R. Schank and B. Nash-Weber). ACL, Arlington, Virginia (1975).
29. R. C. Schank, The primitive ACTs of conceptual dependency. In *Theoretical Issues in Natural Language Processing* (Edited by R. Schank and B. Nash-Webber). ACL, Arlington, Virginia.
30. C. Rieger, The commonsense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. In *Theoretical Issues in Natural Language Processing* (Edited by R. Schank and B. Nash-Webber), pp. 180–195. ACL, Arlington, Virginia (1975).
31. D. L. Waltz, Event shape diagrams. *Proc. 2nd Ann. Nat. Conf. Artificial Intelligence*, Pittsburgh, Penn., August 1981.
32. D. Genter, Talk presented to the *Conf. Cognitive Sci. Soc.*, Yale University, New Haven, Conn., June 1980.
33. N. Cercone, A note on representing adjectives and adverbs. In *Proc. IJCAI*-77, M.I.T., Cambridge, Mass., pp. 139–140, August 1977.
34. K. D. Forbus, Qualitative process theory. *A.I. Memo* 664, M.I.T. A.I. Laboratory, Cambridge, Mass. (1982).
35. C. R. Perrault and P. R. Cohen, It's for your own good: a note on inaccurate reference. In *Elements of Discourse Understanding* (Edited by Joshi, Sag and Webber), pp. 217–230, Cambridge University Press, Cambridge, Mass. (1981).
36. M. Selfridge, A process model of language acquisition. Yale Computer Science Res. Rep 172, Ph.D. dissertation, Yale University, New Haven Conn. (1980).
37. R. Fikes, P. Hart, and N. Nilsson, Learning and executing generalized robot plans, *Artificial Intell.* 3(3), (1972).
38. J. Kolodner, Retrieval and organizational strategies in conceptual memory: a computer model, Yale Computer Science Res. Rep. 187, Ph.D. dissertation, Yale University, New Haven, Conn. (1980).